Bayesian Analysis (2014) **9**, Number 2, pp. 397–424

Bayesian Adaptive Smoothing Splines Using Stochastic Differential Equations

Yu Ryan Yue [*] , Daniel Simpson [†] , Finn Lindgren [‡] and Håvard Rue [§]

Abstract. The smoothing spline is one of the most popular curve-fitting methods, partly because of empirical evidence supporting its effectiveness and partly because of its elegant mathematical formulation. However, there are two obstacles that restrict the use of the smoothing spline in practical statistical work. Firstly, it becomes computationally prohibitive for large data sets because the number of basis functions roughly equals the sample size. Secondly, its global smoothing parameter can only provide a constant amount of smoothing, which often results in poor performances when estimating inhomogeneous functions. In this work, we introduce a class of adaptive smoothing spline models that is derived by solving certain stochastic differential equations with finite element methods. The solution extends the smoothing parameter to a continuous data-driven function, which is able to capture the change of the smoothness of the underlying process. The new model is Markovian, which makes Bayesian computation fast. A simulation study and real data example are presented to demonstrate the effectiveness of our method.

Keywords: Adaptive smoothing, Markov chain Monte Carlo, Smoothing spline, Stochastic differential equation

1 Introduction

The smoothing spline (e.g., Wahba 1990) has been one of the most popular curve-fitting methods, partly because of empirical evidence supporting its effectiveness and partly because of its elegant mathematical formulation. Consider a simple nonparametric regression model:

$$y_i = f(s_i) + \varepsilon_i, \quad i = 1, \ldots, n; \quad s_i \in \mathbb{R}, \tag{1}$$

where (y_i, s_i) are n observations, f is an unknown "smooth" function, and $\varepsilon_i \overset{iid}{\sim} N(0, \tau^{-1})$ with precision (inverse of variance) τ. The smoothing spline estimator of f can be defined as the solution to the following minimization problem:

$$\hat{f} = \arg\min_f \left[\sum_{i=1}^{n} \Big(y_i - f(s_i) \Big)^2 + \lambda \int f^{(p)}(s)^2 ds \right], \tag{2}$$

[*]Baruch College, The City University of New York yu.yue@baruch.cuny.edu
[†]Norwegian University of Science and Technology daniel.simpson@math.ntnu.no
[‡]University of Bath f.lindgren@bath.ac.uk
[§]Norwegian University of Science and Technology Havard.Rue@math.ntnu.no

 DOI:10.1214/13-BA866

where $\lambda > 0$ is the *smoothing parameter* and $f^{(p)}$ is the pth derivative of f. The parameter λ controls the trade-off between fidelity to the data in terms of the residual sum of squares against smoothness of the fit in terms of the integrated squared derivative. The value of p is often taken to be 1 or 2, corresponding to linear and cubic smoothing splines, respectively. From a frequentist point of view, the solution $\hat{f}$ can be explicitly derived within a reproducing kernel Hilbert space and λ is usually estimated via cross-validation or generalized cross-validation (see e.g., Wahba 1990; Gu 2002). From a Bayesian point of view, $\hat{f}$ is the mean of the posterior distribution of f yielded by taking a partially improper Gaussian prior on the function space (Wahba 1990; Eubank 1999; Speckman and Sun 2003).

Despite its popularity, the smoothing spline is well known to perform badly when estimating highly varying functions that have peaks, jumps or frequent curvature transitions (e.g., Eubank 1999). This significant drawback stems from its single smoothing parameter λ, which applies a constant amount of smoothing to the mean function $f(s)$. Consequently, many authors have proposed to make smoothing splines adaptive. Typical methods include local bandwidth selection for the kernel estimator (Staniswalis 1989; Staniswalis and Yandell 1992), local generalized cross-validation (Cummins et al. 2001), adaptive L-splines (Abramovich and Steinberg 1996), hybrid adaptive splines (Luo and Wahba 1997), and spatially adaptive smoothing spline (Pintore et al. 2006). Unfortunately, those methods are computationally intensive and lack flexibility, preventing their implementation in more complicated models (e.g., generalized additive models or bivariate models).

There has been an extensive literature on the adaptive penalized regression spline (P-spline). As an alternative to the smoothing spline, the P-spline uses the basis found by solving the spline smoothing problem for a small representative data set to construct a model for the full data set of interest. The model is typically fitted as a linear mixed model with a roughness penalty on the basis. The covariate points used to obtain the reduced basis are known as the '*knots*' of the spline. To achieve spatial adaptation in the P-spline, one may impose a functional structure on the variances of random effects in the ordinary model. The choice of this smoothing function is often another layer of P-spline functions with a set of *subknots*. Interestingly, there are two versions of P-spline: Eilers and Marx (1996) introduced a P-spline with a B-spline basis and differencing penalty, while Ruppert and Carroll (2000) proposed another P-spline with truncated power basis and ridge penalty. The former was further extended by Lang and Brezger (2004), Brezger and Lang (2006) and Scheipl and Kneib (2009), and the latter was followed by Ruppert et al. (2003), Baladandayuthapani et al. (2005), Crainiceanu et al. (2007) and Krivobokova et al. (2008). O'Sullivan (1986) also introduced a P-spline based on the B-spline basis, but with a more complicated penalty derived from the integrated squared derivative of the fitted curve. Wand and Ormerod (2008) showed that the O'Sullivan spline possesses attractive features, and Simpson et al. (2012) characterized the connection between O'Sullivan splines, classical smoothing splines and the Markovian models considered in this paper. The P-spline models have recently gained incredible popularity due to their easy implementation in sophisticated models (e.g., semiparametric models) and straightforward extensions to

adaptive and higher dimensional smoothing. Unfortunately, the number and location of the knots (and subknots) have a marked effect on the performance of those methods. Actually, choosing appropriate knots for P-spline models is still an open question (Eilers and Marx 2010)

In this work, we propose a unified Bayesian approach to build adaptive smoothing spline models. The method is based on finding Gaussian Markov random field (GMRF) solutions to a class of stochastic differential equations for adaptive smoothing splines. We here provide a brief introduction for GMRF. A random vector $\boldsymbol{w} = (w_1, \ldots, w_n)'$ is a GMRF if its density is

$$[\boldsymbol{w} \mid \delta] \propto |\delta \boldsymbol{Q}|_+^{1/2} \exp\left(-\frac{\delta}{2}(\boldsymbol{w}-\boldsymbol{\mu})'\boldsymbol{Q}(\boldsymbol{w}-\boldsymbol{\mu})\right), \tag{3}$$

where $\delta > 0$ is scale parameter, $\boldsymbol{\mu}$ is mean vector, and $\boldsymbol{Q}$ is a so-called *precision* matrix. The notation $|\boldsymbol{A}|_+$ denotes the generalized determinant of matrix $\boldsymbol{A}$, which is the product of its nonzero eigenvalues. The full conditionals $\pi(w_i \mid \boldsymbol{w}_{-i})$, $i = 1, \ldots, n$, only depend on a set of neighbors $\mathcal{N}_i$ to each site i. The computational gain comes from the fact that the zero-pattern of matrix $\boldsymbol{Q}$ relates directly to the notion of neighbors: $Q_{ij} \neq 0$ if and only if $i \in \mathcal{N}_j \cup j$ (see e.g., Rue and Held 2005, Sec 2.2). The GMRFs allow for fast direct numerical algorithms, as numerical factorization of $\boldsymbol{Q}$ can be done using sparse matrix algorithms at a typical cost of $\mathcal{O}(n)$; see Rue and Held (2005) for detailed algorithms. Such good computational properties are of major importance in Bayesian inferential methods. This is further enhanced by the link to nested integrated Laplace approximations (INLA) (Rue et al. 2009), which allows for fast and accurate Bayesian inference for latent Gaussian field models.

The connection between GMRF and smoothing splines has been explored by several authors. Speckman and Sun (2003) showed that the random walk (RW) models (a subclass of GMRF) (e.g., Fahrmeir and Wagenpfeil 1996; Fahrmeir and Knorr-Held 2000; Fahrmeir and Lang 2001) can be used as priors to derive the discretized Bayesian smoothing spline estimator. Lang et al. (2002) and Yue et al. (2012) made the RW models spatially adaptive by introducing local smoothing parameters into the models. However, all the RW models mentioned above are only appropriate for data observed at regular locations. Lindgren and Rue (2008) considered a second-order RW (RW2) model as a discretely-observed continuous time process, which is derived by solving a stochastic differential equation (SDE) with the finite element method. The resulting RW2 model offers a GMRF representation of the cubic smoothing spline, which has a consistent resolution, equally good performance and more computational efficiency.

The aim of this paper is to extend Lindgren and Rue's model in regard to its spatial adaptation. More specifically, we enable their RW2 model to be spatially adaptive by carefully adding a smoothing function to the SDE. The smoothing function provides a varying amount of smoothing driven by the data. The solution of this modified SDE is thus an adaptive smoothing spline, whose GMRF representation is explicitly available for any collection of locations. Compared to the existing methods, the proposed adaptive smoothing approach has the following advantages. First, it has a convenient computational form, which allows for fast computation and easy implementation in highly

flexible models (e.g., generalized additive models). Second, it has a well-understood continuous limit, providing the comfort that issues like knot choice only have a minimal effect on the model (see Simpson et al. 2012, for a discussion). Third, it inherits nice theoretical and numerical properties from the model in Lindgren and Rue (2008) because it is an intuitive extension of their method to adaptive smoothing.

2 Bayesian smoothing splines using the SDE method

Kimeldorf and Wahba (1970) and Wahba (1978) showed that the smoothing spline $\hat{f}$ in (2) is equivalent to Bayesian estimation with a partially improper prior generated by the following stochastic differential equation (SDE)

$$d^p f(s)/ds^p = \delta^{-\frac{1}{2}} dW(s)/ds, \tag{4}$$

where $\delta > 0$ is scale parameter, $W(s)$ is a zero mean Wiener process with variance s, and $dW(s)/ds$ is "white noise". Letting $(s)_+ = s$ for $s \geq 0$ and $(s)_+ = 0$ otherwise, the exact solution of (4) is

$$f(s) = \beta_0 + \beta_1 s + \cdots + \beta_{p-1} s^{p-1} + \delta^{-\frac{1}{2}} Z(s), \quad s \in \mathbb{R}, \tag{5}$$

where $\beta_0, \beta_1, \ldots, \beta_{p-1} \sim N(0, \xi)$ as $\xi \to \infty$, and $Z(s)$ is a zero mean Gaussian stochastic process with $E[Z(s)Z(t)] = \Sigma(s,t)$ and

$$\Sigma(s,t) = \int_0^1 \frac{(s-u)_+}{(p-1)!} \frac{(t-u)_+}{(p-1)!} du.$$

From a Bayesian point of view, we take a partially improper prior on f, which is "diffuse" on the coefficients of the polynomials of degree $p-1$, and "proper" over the random process $Z(t)$. Given model (1), the solution $\hat{f}(s) = \lim_{\xi \to \infty} E_\xi\{f(s) \,|\, \boldsymbol{y}, \tau, \delta\}$, that is the expectation over the posterior distribution of $f(s)$ with the prior defined in (5). Note that the smoothing parameter $\lambda = \delta/\tau$ now. After taking sensible priors on τ and δ, fully Bayesian inference on $\hat{f}$ can be straightforwardly carried out by Monte Carlo Markov chain (MCMC) (see e.g., Speckman and Sun 2003; Yue et al. 2012)

Unfortunately, the solution (5) does not have any direct Markov properties because its precision matrix $\boldsymbol{Q}$ is completely dense (Rue and Held 2005, chapter 2). It is worthy to note that the solution, however, does have a Markov property on an augmented space, where the derivatives at $\{s_i\}$ are included. Wecker and Ansley (1983) showed how to derive the GMRF of this vector with $2n$ elements, but the computations take about 9/2 the time as for the RW2 model for regular locations (see Rue and Held 2005, chapter 3.5 for details).

To derive a more computationally efficient solution, we solve SDE (4) using the finite element method introduced in Lindgren and Rue (2008). Let $s_1 < s_2 < \cdots < s_n$ be the set of fixed points, which are often observation locations, but do not have to be. We

then construct a finite element representation of $f(s)$ as

$$f(s) \approx \sum_{i=1}^{n} \psi_i(s) w_i, \tag{6}$$

for some chosen basis functions ψ_i and random weights w_i. Letting $h_i = s_{i+1} - s_i$ for $i = 1, \ldots, n-1$, a common choice of basis is the piecewise linear functions

$$\psi_i(s) = \begin{cases} 0, & s < s_{i-1}, \\ \frac{1}{h_{i-1}}(s - s_{i-1}), & s_{i-1} \le s < s_i, \\ 1 - \frac{1}{h_i}(s - s_i), & s_i \le s < s_{i+1}, \\ 0, & s_{i+1} \le s. \end{cases}$$

An interpretation of (6) with these chosen basis functions is that the weights determine the values of the field at the locations, and the values in the interior of the intervals are decided by linear interpolation. The full distribution of the continuously indexed solution is determined by the joint distribution of the weights $\boldsymbol{w} = (w_1, \ldots, w_n)^T$.

Define the inner product $\langle f, g \rangle = \int f(s) g(s) ds$, where the integral is over the region of interest. We then seek a weak solution of (4) satisfying

$$\left\langle \phi, d^2 f / ds^2 \right\rangle \stackrel{d}{=} \delta^{-\frac{1}{2}} \left\langle \phi, dW/ds \right\rangle \tag{7}$$

for *any* sensible test function $\phi(s)$, where $\stackrel{d}{=}$ denotes equality in distribution (Walsh 1986). Note that we only consider the cubic smoothing spline ($p = 2$), which is well known to provide the best overall performance (e.g., Green and Silverman 1994). Since it is impossible to test (7) against every function $\phi(s)$, we choose a finite set $\{\phi_i(s)\}_{i=1}^n$ instead. Moreover, we let the test functions be the same as our basis functions, which is known as the *Galerkin* finite element method. Substituting (6) into (7) with this set of test functions, we end up with a system of linear equations

$$\sum_{i=1}^{n} w_i \left\langle \psi_j, d^2 \psi_i / ds^2 \right\rangle \stackrel{d}{=} \delta^{-\frac{1}{2}} \left\langle \psi_j, dW/ds \right\rangle, \quad j = 1, \ldots, n. \tag{8}$$

It can be shown that the left hand side of (8) can be written as $\boldsymbol{H}\boldsymbol{w}$, where $\boldsymbol{H}$ is an $n \times n$ tridiagonal matrix whose non-zero entries are

$$\boldsymbol{H}[i, i-1] = \frac{1}{h_{i-1}}, \ \boldsymbol{H}[i, i] = -\left(\frac{1}{h_{i-1}} + \frac{1}{h_i} \right), \ \boldsymbol{H}[i, i+1] = \frac{1}{h_i} \tag{9}$$

for $2 \le i \le n-1$, because ψ_i only overlap for neighboring basis functions. The entries of the first and last row in $\boldsymbol{H}$ are zeroes. Given the statistical properties of white noise, the inner product on the right-hand side of (8) is a Gaussian distribution with zero mean and covariance matrix $\boldsymbol{B} = [\langle \psi_i, \psi_j \rangle]_{i,j=1}^n$, whose nonzero entries are given by

$$\boldsymbol{B}[i, i-1] = \frac{h_{i-1}}{6}, \quad \boldsymbol{B}[i, i] = \frac{h_{i-1} + h_i}{3}, \quad \boldsymbol{B}[i, i+1] = \frac{h_i}{6},$$

with modifications at the boundaries. To achieve distribution equality in (8), the random vector $\boldsymbol{w}$ has the density of form (3) with $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{Q} = \boldsymbol{H}'\boldsymbol{B}^{-1}\boldsymbol{H}$. However, this $\boldsymbol{Q}$ is a dense matrix due to dense $\boldsymbol{B}^{-1}$, making the model computationally expensive. To obtain a GMRF with sparse $\boldsymbol{Q}$, we replace $\boldsymbol{B}$ by a diagonal matrix $\tilde{\boldsymbol{B}}$ with $\tilde{\boldsymbol{B}}[i,i] = \langle \psi_i, 1 \rangle$, that is

$$\tilde{\boldsymbol{B}}[1,1] = \frac{h_1}{2}, \quad \tilde{\boldsymbol{B}}[i,i] = \frac{h_{i-1} + h_i}{2}, \quad \tilde{\boldsymbol{B}}[n,n] = \frac{h_{n-1}}{2}. \tag{10}$$

Lindgren and Rue (2008) showed that such a modification does not change the solution. It is straightforward to verify that $\boldsymbol{Q}$ has rank $n-2$, with the null space spanned by vectors $(1,\ldots,1)^T$ and $(s_1,\ldots,s_n)^T$. It indicates that the resulting field is invariant to the addition of a linear trend, coinciding with the result obtained by Wahba (1978) for cubic smoothing splines.

As weights of the basis expansion in (6), the derived GMRF approximates the continuous function f everywhere. Simpson et al. (2012) showed that the convergence of this approximation depends solely on the basis functions. Given any set of enough s_i, using the piecewise linear functions yields the best finite approximation regardless of their locations. Also, the SDE method works for any set of test and basis functions when all of the computations make sense. Actually, Simpson et al. showed that O'Sullivan splines can be exactly derived by solving the SDE in (4) using cubic B-splines as basis functions and their second derivatives as test functions. However, one should be aware that the wrong choice of global basis functions will destroy the Markov structure, and not all sets of basis functions will provide good approximations to f.

3 Bayesian adaptive smoothing splines

One big advantage of using the SDE method to model smoothing splines is its straightforward extension to adaptive smoothing. The basic idea is to replace the single smoothing parameter by a smoothing function varying in space. This will make us control local smoothing properties of the spline function. We here present two different adaptive SDE formulations, from both of which we are able to derive the GMRF models that provide appropriate adaptive smoothing.

3.1 Adaptive SDE I

One way to extend SDE (4) is as follows:

$$\lambda(s) d^2 f(s)/ds^2 = dW(s)/ds, \tag{11}$$

where the positive $\lambda(s)$ can be viewed as an adaptive smoothing function, compared to the global smoothing parameter λ in ordinary smoothing splines. A small $\lambda(s)$ allows a big second derivative of $f(s)$ for roughness, while a large value diminishes the derivative to increase smoothness. The solution to (11) is related to the spline model introduced

in Pintore et al. (2006), minimizing

$$\sum_{i=1}^{n}\Big(y_i - f(s_i)\Big)^2 + \int \left[\lambda(s)f''(s)\right]^2 ds. \tag{12}$$

Using a piecewise-constant model for $\lambda(s)$, Pintore et al. derived closed-form solutions for the corresponding reproducing kernels of the Hilbert space. Their method, however, is computationally intensive since the matrix of reproducing kernels is completely dense.

Following the non-adaptive case, we seek a weak solution of (11) by achieving

$$\left\langle \psi_j, \lambda d^2 f/ds^2 \right\rangle \stackrel{d}{=} \left\langle \psi_j, dW/ds \right\rangle, \quad j = 1, \ldots, n. \tag{13}$$

Using the basis representation in (6) as well as Galerkin approximation, the left hand side of (13) can be proved to be $\boldsymbol{\Lambda H w}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix with $(\lambda(s_1), \ldots, \lambda(s_n))$ on the diagonal and $\boldsymbol{H}$ is the matrix as in (9) (see Appendix for the proof). Since the right-hand side of (13) is the same as that of (8), the $\boldsymbol{w}$ is also a GMRF with zero mean and precision matrix

$$\boldsymbol{Q}_\lambda = \boldsymbol{H}'\boldsymbol{\Lambda}\tilde{\boldsymbol{B}}^{-1}\boldsymbol{\Lambda H}.$$

It is easy to see that $\boldsymbol{Q}_\lambda$ is symmetric and banded with non-zero entries of the ith row given by

$$\begin{aligned}
\boldsymbol{Q}_\lambda[i, i-2] &= \frac{2\lambda^2(s_{i-1})}{h_{i-2}h_{i-1}(h_{i-2}+h_{i-1})}, \quad \boldsymbol{Q}_\lambda[i, i-1] = -\frac{2}{h_{i-1}^2}\left(\frac{\lambda^2(s_{i-1})}{h_{i-2}} + \frac{\lambda^2(s_i)}{h_i}\right), \\
\boldsymbol{Q}_\lambda[i, i] &= \frac{2\lambda^2(s_{i-1})}{h_{i-1}^2(h_{i-2}+h_{i-1})} + \frac{2\lambda^2(s_i)}{h_{i-1}h_i}\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right) + \frac{2\lambda^2(s_{i+1})}{h_i^2(h_i+h_{i+1})}.
\end{aligned}$$

At the discretization boundaries, we use the convention that terms with non-existing components are ignored, that is $h_{-1} = h_0 = h_n = h_{n+1} = \infty$. This affects only the upper left and lower right corner of $\boldsymbol{Q}_\lambda$ as follows:

$$\begin{aligned}
\boldsymbol{Q}_\lambda[1,1] &= \frac{2\lambda^2(s_2)}{h_1^2(h_1+h_2)}, \quad \boldsymbol{Q}_\lambda[2,1] = -\frac{2\lambda^2(s_2)}{h_1^2 h_2}, \\
\boldsymbol{Q}_\lambda[2,2] &= \frac{2\lambda^2(s_3)}{h_2^2(h_2+h_3)} + \frac{2\lambda^2(s_2)}{h_1 h_2}\left(\frac{1}{h_1} + \frac{1}{h_2}\right), \\
\boldsymbol{Q}_\lambda[n-1,n-1] &= \frac{2\lambda^2(s_{n-2})}{h_{n-2}^2(h_{n-3}+h_{n-2})} + \frac{2\lambda^2(s_{n-1})}{h_{n-2}h_{n-1}}\left(\frac{1}{h_{n-2}} + \frac{1}{h_{n-1}}\right), \\
\boldsymbol{Q}_\lambda[n,n] &= \frac{2\lambda^2(s_{n-1})}{h_{n-1}^2(h_{n-2}+h_{n-1})}, \quad \boldsymbol{Q}_\lambda[n,n-1] = -\frac{2\lambda^2(s_{n-1})}{h_{n-2}h_{n-1}^2}.
\end{aligned}$$

Note that $\boldsymbol{Q}_\lambda$ does not involve $\lambda(s_1)$ or $\lambda(s_n)$ because the first and last rows of $\boldsymbol{H}$ are zeroes.

3.2 Adaptive SDE II

An alternative SDE that we can use for adaptive smoothing splines is

$$d^2\lambda(s)f(s)/ds^2 = dW(s)/ds, \tag{14}$$

where $\lambda(s)$ can be seen as an instantaneous variance or local scaling, which compresses and stretches the function. A small $\lambda(s)$ compresses the scale giving quick oscillations, while a high value stretches $f(s)$, decreasing the roughness. Adopting the notation $\tilde{f}(s) = \lambda(s)f(s)$, formulation (14) corresponds to minimizing

$$\sum_{i=1}^{n}\left(y_i - \tilde{f}(s_i)\right)^2 + \int \tilde{f}''(s)^2 ds.$$

The weak solution of (14) can also be found using the Galerkin method to satisfy

$$\left\langle \psi_j, d^2\lambda f/ds^2 \right\rangle \stackrel{d}{=} \left\langle \psi_j, dW/ds \right\rangle, \quad j = 1, \ldots, n, \tag{15}$$

whose left-hand side can be written as $\boldsymbol{H}\boldsymbol{\Lambda}\boldsymbol{w}$, where $\boldsymbol{H}$ and $\boldsymbol{\Lambda}$ are defined as above (see Appendix for the proof). Again, the $\boldsymbol{w}$ is a GMRF with zero mean and precision matrix

$$\boldsymbol{Q}_\lambda = \boldsymbol{\Lambda}\boldsymbol{H}'\tilde{\boldsymbol{B}}^{-1}\boldsymbol{H}\boldsymbol{\Lambda},$$

whose nonzero entries can be explicitly written out as

$$\boldsymbol{Q}_\lambda[i, i-2] = \frac{2\lambda(s_{i-2})\lambda(s_i)}{h_{i-2}h_{i-1}(h_{i-2}+h_{i-1})}, \; \boldsymbol{Q}_\lambda[i, i-1] = \frac{2\lambda(s_{i-1})\lambda(s_i)}{-h_{i-1}^2}\left(\frac{1}{h_{i-2}} + \frac{1}{h_i}\right),$$
$$\boldsymbol{Q}_\lambda[i, i] = \frac{2\lambda^2(s_i)}{h_{i-1}^2(h_{i-2}+h_{i-1})} + \frac{2\lambda^2(s_i)}{h_{i-1}h_i}\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right) + \frac{2\lambda^2(s_i)}{h_i^2(h_i+h_{i+1})},$$

with corrected boundary entries

$$\boldsymbol{Q}_\lambda[1,1] = \frac{2\lambda^2(s_1)}{h_1^2(h_1+h_2)}, \quad \boldsymbol{Q}_\lambda[2,1] = -\frac{2\lambda(s_1)\lambda(s_2)}{h_1^2 h_2},$$
$$\boldsymbol{Q}_\lambda[2,2] = \frac{2\lambda^2(s_2)}{h_2^2(h_2+h_3)} + \frac{2\lambda^2(s_2)}{h_1 h_2}\left(\frac{1}{h_1} + \frac{1}{h_2}\right),$$
$$\boldsymbol{Q}_\lambda[n-1,n-1] = \frac{2\lambda^2(s_{n-1})}{h_{n-2}^2(h_{n-3}+h_{n-2})} + \frac{2\lambda^2(s_{n-1})}{h_{n-2}h_{n-1}}\left(\frac{1}{h_{n-2}} + \frac{1}{h_{n-1}}\right),$$
$$\boldsymbol{Q}_\lambda[n,n] = \frac{2\lambda^2(s_n)}{h_{n-1}^2(h_{n-2}+h_{n-1})}, \quad \boldsymbol{Q}_\lambda[n,n-1] = -\frac{2\lambda(s_{n-1})\lambda(s_n)}{h_{n-2}h_{n-1}^2}.$$

3.3 Modeling the adaptive smoothing function

To implement a fully Bayesian inference, we need a prior taken on the smoothing function $\lambda(s)$, which is assumed to be continuous and differentiable. Yue and Speckman (2010) and Yue et al. (2012) have proved that the prior on $\lambda(s)$ must be proper in order to guarantee a proper posterior for such adaptive smoothing models. Since it is restricted to be positive, we model $\lambda(s)$ on its log scale: $\nu(s) = \log(\lambda(s))$. There are several ways to model $\nu(s)$ as a smooth function. One may follow the basis expansion in (6) and let $\nu(s) = \sum_{k=1}^{m} \omega_k(s)\gamma_k$, a weighted sum of B-spline basis functions $\omega_k(s)$ at knots $s'_1, s'_2, \ldots, s'_m$. The random weights $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_m)'$ are assumed to follow a multivariate normal distribution with mean zero and precision matrix $\boldsymbol{R}$. To ensure the propriety of the posterior distribution, $\boldsymbol{R}$ must be a positive definite matrix. One simple choice is $\boldsymbol{R} = \eta \boldsymbol{I}$, where η is a fixed small value. It is equivalent to taking independent and diffuse Gaussian priors on γ_k's. Unfortunately, this curve fitting method can be sensitive to the number and locations of the knots. To relieve the issue, we present a more sophisticated SDE approach as follows.

Lindgren et al. (2011) derived an explicit link between GMRF and Matérn Gaussian fields by solving certain stochastic partial differential equations. We here apply their idea to our univariate case by considering the SDE

$$\left(\kappa^2 - d^2/ds^2\right) \nu(s) = \eta^{-\frac{1}{2}} dW(s)/ds, \tag{16}$$

where $\kappa > 0$ is fixed and $\eta > 0$ is scale parameter. Again, we use the Galerkin method to weakly solve (16) as a linear equation system,

$$\sum_{k=1}^{m} \gamma_k \left\langle \omega_\ell, \left(\kappa^2 - d^2 ds^2\right) \omega_k \right\rangle \stackrel{d}{=} \eta^{-\frac{1}{2}} \left\langle \omega_\ell, dW/ds \right\rangle, \quad \ell = 1, \ldots, m. \tag{17}$$

With a piecewise linear basis, it can be shown that the left hand side of (17) is $(\kappa \boldsymbol{B} - \boldsymbol{H})\boldsymbol{\gamma}$ and the right hand side is the Gaussian random vector as in the previous SDEs. As a result, the precision matrix of the corresponding GMRF is given by

$$\boldsymbol{R} = (\kappa^2 \boldsymbol{B} - \boldsymbol{H})' \boldsymbol{B}^{-1} (\kappa^2 \boldsymbol{B} - \boldsymbol{H}) = \kappa^4 \boldsymbol{B} - \kappa^2 (\boldsymbol{H}' + \boldsymbol{H}) + \boldsymbol{H}' \boldsymbol{B}^{-1} \boldsymbol{H}.$$

To make $\boldsymbol{R}$ sparse, we replace $\boldsymbol{B}$ by diagonal $\tilde{\boldsymbol{B}}$ as before. Note that $\boldsymbol{R}$ is positive definite and it gets singular as κ goes to zero. This GMRF is a proper prior with nice computational properties. More importantly, it makes the fit of the adaptive smoothing function robust to the knots as long as there are enough of them.

4 Bivariate adaptive smoothing

Two-dimensional thin-plate splines (TPS) (e.g., Duchon 1977; Gu 2002) are a natural bivariate extension of cubic smoothing splines. The TPS estimator can be obtained by solving

$$\hat{f} = \arg\min_{f} \left[\sum_{i=1}^{n} \left(y_i - f(\boldsymbol{s}_i) \right)^2 + \lambda J_2(f) \right], \quad \boldsymbol{s}_i \in \mathbb{R}^2$$

where penalty function $J_2(f)$ is of form

$$J_2(f) = \iint \left[\left(\frac{\partial^2 f(\boldsymbol{s})}{\partial s_1^2} \right)^2 + 2 \left(\frac{\partial^2 f(\boldsymbol{s})}{\partial s_1 \partial s_2} \right)^2 + \left(\frac{\partial^2 f(\boldsymbol{s})}{\partial s_2^2} \right)^2 \right] ds_1\, ds_2. \tag{18}$$

Assuming that the integrable derivatives of f vanish at infinity, the TPS penalty in (18) can be shown to be $\tilde{J}_2(f) = \iint [\Delta f(\boldsymbol{s})]^2\, ds_1\, ds_2$, where $\Delta = \left(\partial^2/\partial s_1^2 + \partial^2/\partial s_2^2\right)$ is the two-dimensional Laplacian operator (Wahba 1990; Yue and Speckman 2010). Such a form of penalty inspires us to solve the following SDE

$$\Delta f(\boldsymbol{s}) = dW(\boldsymbol{s})/d\boldsymbol{s}, \tag{19}$$

where $dW(\boldsymbol{s})/d\boldsymbol{s}$ is spatial Gaussian white noise, to obtain a TPS estimator. Yue and Speckman (2010) constructed an efficient GMRF representation of the solution to (19) on a regular grid. It is, however, not yet fully practical since one often wants to avoid interpolating the locations of observations to the nearest grid-point, and to allow for finer resolution where details are required. Following Lindgren et al. (2011), we use the finite element method to find a weak solution to (19) on an irregular grid. It is a direct generalization of the SDE approach in the univariate case. We briefly describe the procedure and technical details can be found in Lindgren et al. (2011).

We first construct a triangulation by subdividing $\mathbb{R}^2$ into a set of non-intersecting triangles, where any two triangles meet in at most a common edge or corner. In most cases we place initial vertices at the locations for the observations, and add additional vertices to satisfy overall soft constraints of the triangles. Letting $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ be the vertices of the triangulation Ω, we then represent a smooth bivariate function $f(\boldsymbol{s})$ using piecewise linear basis functions defined over Ω, i.e., $f(\boldsymbol{s}) = \sum_{i=1}^n \psi_i(\boldsymbol{s}) w_i$. The basis function ψ_i is equal to one at the ith vertex and is zero at all other vertices. As in the univariate case, we solve a weak formulation of (19) using the finite element method to obtain the distribution of weights $\boldsymbol{w} = (w_1, \ldots, w_n)'$. It turns out that $\boldsymbol{w}$ is Gaussian with mean zero and precision matrix $\boldsymbol{Q} = \boldsymbol{G}'\boldsymbol{C}^{-1}\boldsymbol{G}$, where

$$\boldsymbol{G}[i,j] = \int_\Omega \nabla\psi_i(\boldsymbol{s}) \cdot \nabla\psi_j(\boldsymbol{s}) d\boldsymbol{s} \quad \text{and} \quad \boldsymbol{C}[i,j] = \int_\Omega \psi_i(\boldsymbol{s})\psi_j(\boldsymbol{s}) d\boldsymbol{s}.$$

Due to the local nature of basis functions, $\boldsymbol{G}$ and $\boldsymbol{C}$ are highly sparse matrices. However, $\boldsymbol{w}$ is not a GMRF since $\boldsymbol{C}^{-1}$ is a dense matrix. Fortunately, we can replace $\boldsymbol{C}$ by a diagonal matrix $\tilde{\boldsymbol{C}} = \text{diag}\left[\int_\Omega \psi_i(\boldsymbol{s}) d\boldsymbol{s}\right]$, which gives essentially the same numerical result and can be shown not to increase the rate of convergence of the approximation (Appendix C.5 of Lindgren et al. 2011). Now $\boldsymbol{w}$ becomes a GMRF with precision matrix $\boldsymbol{Q} = \boldsymbol{G}'\tilde{\boldsymbol{C}}^{-1}\boldsymbol{G}$.

To extend the TPS estimator to be adaptive, we can simply modify (19) by adding the adaptive function $\lambda(\boldsymbol{s})$, i.e., $\Delta\lambda(\boldsymbol{s})f(\boldsymbol{s}) = dW(\boldsymbol{s})/d\boldsymbol{s}$. It is a direct bivariate extension of the adaptive SDE in (14). Similarly, this SDE can be solved using the finite element method and the solution is another GMRF with mean zero and precision matrix $\boldsymbol{Q}_\lambda =$

$\boldsymbol{\Lambda}\boldsymbol{G}'\tilde{\boldsymbol{C}}^{-1}\boldsymbol{G}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix of the values of λ at vertices. Following the univariate case, we model $\log\lambda(\boldsymbol{s})$ with a basis expansion and take independent Gaussian priors on the basis weights.

5 Posterior Inference

Besides their intriguing adaptive smoothing and computational properties, one of the most exciting aspects of the proposed SDE splines is their flexibility: they can be used in a large class of models. In this section, we show how to implement our method in simple nonparametric regression models and more flexible structured additive models.

5.1 Simple nonparametric regresson model

Given model (1), it is easy to build a fully Bayesian hierarchical model for smoothing splines. We take the proposed SDE prior on f and diffuse gamma priors on the precision parameters. Let $\boldsymbol{\Psi} = \{\psi_j(s_i)\}_{i,j=1}^n$ and $\boldsymbol{\Omega} = \{\omega_k(t_\ell)\}_{k,\ell=1}^m$ be matrices of basis functions for $f(s)$ and $\nu(s)$, respectively. Then, the hierarchical model has following structure:

$$
\begin{aligned}
&\boldsymbol{y} = \boldsymbol{\Psi}\boldsymbol{w} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \tau^{-1}\boldsymbol{I}),\\
&[\boldsymbol{w} \,|\, \boldsymbol{\lambda}] \propto |\boldsymbol{Q}_\lambda|_+^{1/2}\exp\left(-\frac{1}{2}\boldsymbol{w}'\boldsymbol{Q}_\lambda\boldsymbol{w}\right),\\
&\log(\lambda_i) = \nu_i, \quad \boldsymbol{\nu} = \boldsymbol{\Omega}\boldsymbol{\gamma},\\
&[\boldsymbol{\gamma} \,|\, \eta] \propto |\eta\boldsymbol{R}|^{1/2}\exp\left(-\frac{\eta}{2}\boldsymbol{\gamma}'\boldsymbol{R}\boldsymbol{\gamma}\right),\\
&\tau, \eta \sim \text{Gamma}(\epsilon, \epsilon),
\end{aligned}
$$

where ξ is a small value, say 0.001. To make inference on this model, we propose a Monte Carlo Markov chain (MCMC) simulation algorithm. We here focus on how to sample $\boldsymbol{\gamma}$ from its full conditional because the remaining samples are straightforward to acquire. The full conditional of $\boldsymbol{\gamma}$ is not a regular density, so we have to employ Metropolis-Hastings sampling. We present an efficient algorithm to sample $\boldsymbol{\gamma}$ when using the first adaptive SDE. Unfortunately, we have not found an equivalently efficient method for the second adaptive SDE, which, however, can be taken care of by the method described in the next section.

A good proposal distribution is the key to the successful Metropolis-Hastings algorithm. It is helpful to see that the GMRF derived from the first adaptive SDE can be written as

$$
[\boldsymbol{w} \,|\, \boldsymbol{\gamma}] \propto \exp\left(\sum_{i=1}^n \left(\gamma_i - e^{\gamma_i}\tilde{w}_i^2\right)\right),
$$

where $\tilde{\boldsymbol{w}} = (0, \tilde{w}_2, \ldots, \tilde{w}_{n-1}, 0)' = \boldsymbol{H}\boldsymbol{w}$ (note the first and last rows of $\boldsymbol{H}$ are zeroes). The full conditional of $\boldsymbol{\gamma}$ is then given by $\mathcal{F}(\boldsymbol{\gamma} \,|\, \boldsymbol{w}, \eta) \propto [\boldsymbol{w} \,|\, \boldsymbol{\gamma}][\boldsymbol{\gamma} \,|\, \eta]$. Following Rue and Held (2005), we construct an accurate GMRF approximation for $\mathcal{F}(\boldsymbol{\gamma} \,|\, \boldsymbol{w}, \eta)$ as follows.

First, we approximate $[\boldsymbol{w} \,|\, \boldsymbol{\gamma}]$ by taking a second-order Taylor expansion on e^{γ_i} at given γ_{0i}, that is

$$[\boldsymbol{w} \,|\, \boldsymbol{\gamma}] \approx \exp\left(\sum_{i=1}^{n}\left(a_i + b_i\gamma_i - \frac{1}{2}c_i\gamma_i^2\right)\right),$$

where a_i is the nuisance parameter, $b_i = 1 - \delta e^{\gamma_{0i}}(1-\gamma_{0i})\tilde{w}_i^2$ and $c_i = \delta e^{\gamma_{0i}}\tilde{w}_i^2$. Letting $\boldsymbol{b} = (b_1, \ldots, b_n)'$ and $\boldsymbol{c} = (c_1, \ldots, c_n)'$, then

$$\begin{aligned}\mathcal{P}(\boldsymbol{\gamma} \,|\, \boldsymbol{\gamma}_0, \boldsymbol{w}, \eta) &\propto \exp\left(-\frac{\eta}{2}\boldsymbol{\gamma}'\boldsymbol{R}\boldsymbol{\gamma} + \sum_{i=1}^{n}\left(a_i + b_i\gamma_i - \frac{1}{2}c_i\gamma_i^2\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\boldsymbol{\gamma}'\Big(\eta\boldsymbol{R} + \mathrm{diag}(\boldsymbol{c})\Big)\boldsymbol{\gamma} + \boldsymbol{b}'\boldsymbol{\gamma}\right),\end{aligned}$$

is a GMRF with sparse precision matrix $\boldsymbol{Q} = \eta\boldsymbol{R} + \mathrm{diag}(\boldsymbol{c})$. The $\mathcal{P}(\boldsymbol{\gamma} \,|\, \boldsymbol{\gamma}_0, \boldsymbol{w}, \eta)$ can be used to approximate $\mathcal{F}(\boldsymbol{\gamma} \,|\, \boldsymbol{w}, \eta)$ and the accuracy of this approximation depends on $\boldsymbol{\gamma}_0$. As suggested by Rue and Held (2005), we choose $\boldsymbol{\gamma}_0$ to be the mode of $\mathcal{F}(\boldsymbol{\gamma} \,|\, \boldsymbol{w}, \eta)$ that can be obtained using, say Newton-Raphson. Using this GMRF approximation as the proposal distribution, we can update the whole $\boldsymbol{\gamma}$ by accepting proposal $\boldsymbol{\gamma}^*$ with probability

$$\min\left(1, \quad \frac{\mathcal{F}(\boldsymbol{\gamma}^* \,|\, \boldsymbol{w}, \eta)\mathcal{P}(\boldsymbol{\gamma} \,|\, \boldsymbol{\gamma}_0, \boldsymbol{w}, \eta)}{\mathcal{F}(\boldsymbol{\gamma} \,|\, \boldsymbol{w}, \eta)\mathcal{P}(\boldsymbol{\gamma}^* \,|\, \boldsymbol{\gamma}_0, \boldsymbol{w}, \eta)}\right).$$

Other full conditionals are given by

$$\begin{aligned}&(\boldsymbol{w} \,|\, \boldsymbol{\gamma}, \tau) \sim N(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w), \quad \boldsymbol{\mu}_w = \tau\boldsymbol{\Sigma}_w\boldsymbol{\Psi}'\boldsymbol{y} \quad \text{and} \quad \boldsymbol{\Sigma}_w = (\tau\boldsymbol{\Psi}'\boldsymbol{\Psi} + \boldsymbol{Q}_\lambda)^{-1} \\ &(\tau \,|\, \boldsymbol{w}) \sim \mathrm{Gamma}(n/2 + \epsilon,\ \|\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{w}\|^2/2 + \epsilon) \\ &(\eta \,|\, \boldsymbol{\gamma}) \sim \mathrm{Gamma}(n/2 - 1 + \epsilon,\ \boldsymbol{\gamma}'\boldsymbol{R}\boldsymbol{\gamma}/2 + \epsilon),\end{aligned}$$

all of which can be easily sampled.

5.2 Structured additive model

We now consider a class of structured additive models, which are fairly flexible and used in a wealth of applications (e.g., Fahrmeir and Tutz 2001). In these models, the response variable y_i is assumed to belong to an exponential family, where the mean μ_i is linked to a structured additive predictor through a link function $g(\cdot)$,

$$g(\mu_i) = \beta_0 + \sum_{j=1}^{n_\beta} x_{ji}\beta_j + \sum_{k=1}^{n_f} f_k(s_{ki}) \quad i = 1, \ldots, n.$$

Here, the β_j's represent the linear effect of covariates $\boldsymbol{x}$ and the f_k's are smooth functions of the covariates $\boldsymbol{s}$ (univariate or bivariate). We then take diffuse Gaussian priors on β_0

and β_j's, and SDE priors on f_k's, to build a *latent Gaussian model*. Let $\boldsymbol{f}$ denote the vector of all the latent Gaussian variables, and $\boldsymbol{\theta}$ the vector of hyper parameters, which are not necessarily Gaussian. It is common to use MCMC sampling to make inference for such models. However, it is well known that MCMC methods tend to exhibit poor performance because first, the components of $\boldsymbol{f}$ are strongly dependent on each other; second, $\boldsymbol{\theta}$ and $\boldsymbol{f}$ are also strongly dependent, especially when n is large (Rue and Held 2005; Rue et al. 2009).

As an alternative to MCMC, Rue et al. (2009) introduced an innovative Bayesian inference tool based on integrated nested Laplace approximations (INLAs). The INLA approach can handle general Gaussian hierarchical models, including the models developed in this paper. It accurately approximates marginal posterior densities and computes estimates much faster than general MCMC techniques. We now briefly review the INLA method and refer readers to Rue et al. (2009) for details. With likelihood $\pi(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{\theta})$ and prior densities $\pi(\boldsymbol{f}|\boldsymbol{\theta})$ and $\pi(\boldsymbol{\theta})$, the joint posterior of our model is given by

$$\pi(\boldsymbol{f},\boldsymbol{\theta}|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{\theta})\pi(\boldsymbol{f}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}).$$

We need to find the posterior marginals $\pi(f_i|\boldsymbol{y})$ and $\pi(\theta_j|\boldsymbol{y})$. The INLA approach is based on the following approximation for the posterior marginal of $\boldsymbol{\theta}$:

$$\tilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y}) \propto \left.\frac{\pi(\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{y})}{\pi_G(\boldsymbol{f}|\boldsymbol{\theta},\boldsymbol{y})}\right|_{\boldsymbol{f}=\boldsymbol{f}^\star(\boldsymbol{\theta})},$$

where $\pi_G(\boldsymbol{f}|\boldsymbol{\theta},\boldsymbol{y})$ is the Gaussian approximation to the full conditional of $\boldsymbol{f}$, and $\boldsymbol{f}^\star(\boldsymbol{\theta})$ is the mode of the full conditional of $\boldsymbol{f}$. The approximated marginals are then constructed as follows:

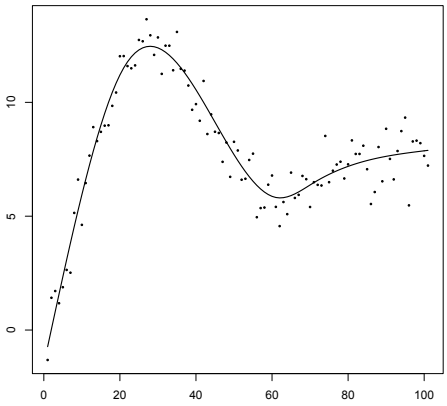
$$\begin{aligned}\tilde{\pi}(\theta_j|\boldsymbol{y}) &= \int \tilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}_{-j},\\ \tilde{\pi}(f_i|\boldsymbol{y}) &= \int \tilde{\pi}(f_i|\boldsymbol{\theta},\boldsymbol{y})\tilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta},\end{aligned}$$

where $\boldsymbol{\theta}_{-j}$ denotes a subvector of $\boldsymbol{\theta}$ without element θ_j. The approximated marginal of θ_j can be obtained by summing out the remaining variables $\boldsymbol{\theta}_{-j}$ from $\tilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$. The approximated marginal of f_i is obtained by, first, approximating the full conditional of f_i with another Laplace approximation:

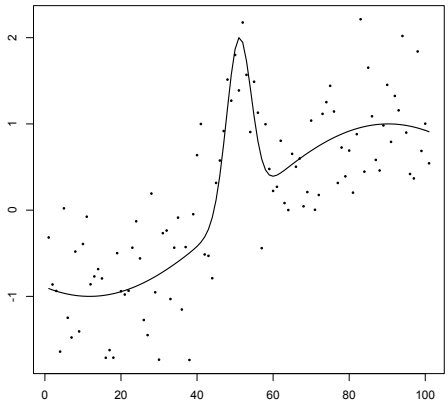
$$\tilde{\pi}(f_i|\boldsymbol{\theta},\boldsymbol{y}) \propto \left.\frac{\pi(\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{y})}{\pi_{GG}(\boldsymbol{f}_{-i}|f_i,\boldsymbol{\theta},\boldsymbol{y})}\right|_{\boldsymbol{f}_{-i}=\boldsymbol{f}^\star_{-i}(f_{-i},\boldsymbol{\theta})},$$

where $\tilde{\pi}_{GG}$ is the Gaussian approximation to $\boldsymbol{f}_{-i}|f_i,\boldsymbol{\theta},\boldsymbol{y}$ and $\boldsymbol{f}^\star_{-i}(f_{-i},\boldsymbol{\theta})$ is the mode configuration. Then, we numerically integrate out the parameters $\boldsymbol{\theta}$ from $\tilde{\pi}(f_i|\boldsymbol{\theta},\boldsymbol{y})$. This nested approach makes the Laplace approximations very accurate. However, the INLA method has a limitation: it only works efficiently when the number of hyperparameters in $\boldsymbol{\theta}$ is small, say less than 15. The reason is that it becomes extremely expensive to numerically integrate out $\boldsymbol{\theta}$ as its dimension increases. As a result, we have to model the adaptive smoothing function $\lambda(s)$ using a B-spline basis with only a few knots if we want to use INLA.

(a) Example 1



(b) Example 2



(c) Example 3

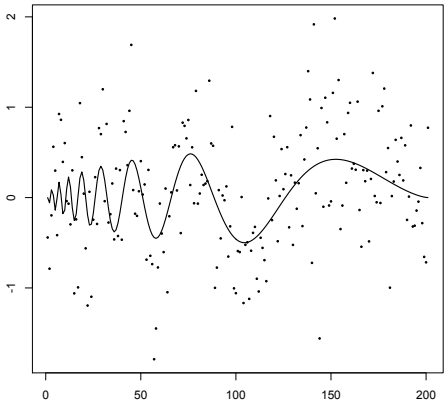


Figure 1: The three true functions used in the simulation study together with one sample.

	Example 1	Example 2	Example 3
SDE-v1	.0598 [.0432, .0834]	.00487 [.00375, 0.0420]	.00705 [.00617, .00844]
SDE-v2	.0597 [.0430, .0835]	.00644 [.00530, .00801]	.00548 [.00461, .00660]
FAPS	.0575 [.0397, .0813]	.00591 [.00449, .00720]	.00875 [.00786, .00968]
OSS	.0600 [.0401, .0853]	.00889 [.00731, .01077]	.00916 [0.0081, .01022]

Table 1: Simulation study. Median MSE with first and third quartiles in brackets from 200 simulated data sets using the SDE, FAPS and OSS procedures. SDE-v1 and SDE-v2 denote the models derived from the first and second adaptive SDEs, respectively.

6 Simulated examples

In this section we consider three mean functions: a slowly-varying smooth function, a function with a sharp peak, that is spatially inhomogeneously smooth, and a highly-oscillating Doppler function. The functions together with simulated data are shown in Figure 1. In Example 1, the true mean function is a B-spline with three internal knots at (0.2, 0.6, 0.7) and coefficients (20, 4, 6, 11, 6). The function is evaluated on a regular grid of $n = 101$ points, and zero-mean Gaussian noise is added to the function with standard deviation 0.9. In Example 2, the true mean function is $f(s) = \sin(s) + 2\exp(-30s^2)$ for $s \in [-2, 2]$, evaluated at $n = 101$ regularly spaced points, and the standard deviation of the Gaussian noise is 0.5. In Example 3, the Doppler mean function is given by $f(s) = \sqrt{s(1-s)}\sin(2\pi(1+\epsilon)/(s+\epsilon))$ for $\epsilon = 0.125$, evaluated at $n = 201$ regularly spaced points, and the standard deviation of the Gaussian noise is 0.2.

We simulate 200 data sets for each true mean function and compare our SDE smoothing spline (SDE) estimates with ordinary smoothing spline (OSS) estimates and fast adaptive P-spline (FAPS) estimates (Krivobokova et al. 2008), using mean squared error

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{f}(s_i) - f(s_i)\right]^2.$$

We use non-adaptive versions of the SDE and FAPS models in the first example and their adaptive versions in the rest of the examples. The first adaptive SDE model (SDE-v1) is fitted by MCMC while the second one (SDE-v2) is estimated by INLA. Note that with MCMC we use the same number of knots as the data points, while with INLA we use $m = 5$ knots. The median of the MSE, together with its first and third quartile, are reported in Table 3. As we can see, the three methods perform similarly when estimating the smooth function in Example 1. In particular, the two SDE models yield very close MSE's, which means INLA makes as accurate inference as MCMC does but with much faster computation. (The two SDE models are identical in the non-adaptive case.) Not surprisingly, the FAPS and SDE outperform OSS in estimating the highly-varying functions in Example 2 and 3. Compared to FAPS, SDE-v1 outperforms in Example 2 and both SDE models outperform in Example 3, indicating that our SDE models provide appropriate adaptive smoothing. It is also interesting to see that the two SDE models yield close but not equal performance. We thus suggest fitting both

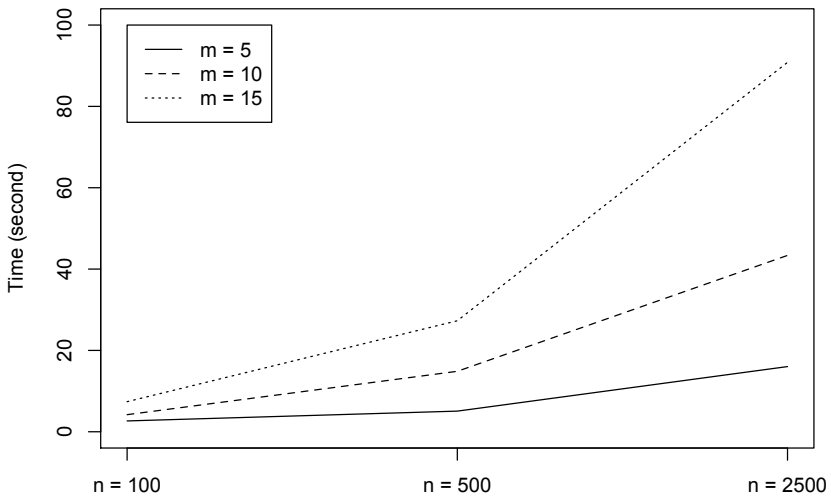


Figure 2: Computational times of INLA in simulation study.

models to a particular data set in practice.

To evaluate the computational performance of INLA method, we show how the computational times change with sample size n and number of knots m in Figure 2. Given a small value of m, INLA is quite fast: it takes about 10 seconds to finish computation for $n = 2,500$. Also, the computational times increase almost linearly with sample sizes. However, INLA significantly slows down as the value of m increases. Regarding the MCMC approach, we run 12,000 iterations with 2,000 burn-in and save samples every two iterations. A couple of trace plots are shown in Figure 3, in which there still exists a little mixing issue. Furthermore, MCMC is much slower than INLA: it takes 16.204 seconds for MCMC to finish 12,000 iterations with $n = 100$ while INLA only needs 2.661 seconds. All the computations are implemented on a standard iMac with 3.06 GHz Intel CPU and 4 GB memory.

7 Mackerel egg survey example

Most commercially exploitable fish stocks in the world have been over exploited. Effective management of stocks rests on sound fish stock assessment, but counting the number of catchable fish of any given species is almost impossible. One solution is to assess the number of eggs produced by a stock, and then to work out the number (or more often mass) of adult fish required to produce this number. Such egg data are obtained by sending out scientific cruise ships to sample eggs at each station of some predefined sampling grid over the area occupied by a stock. The number of eggs of the

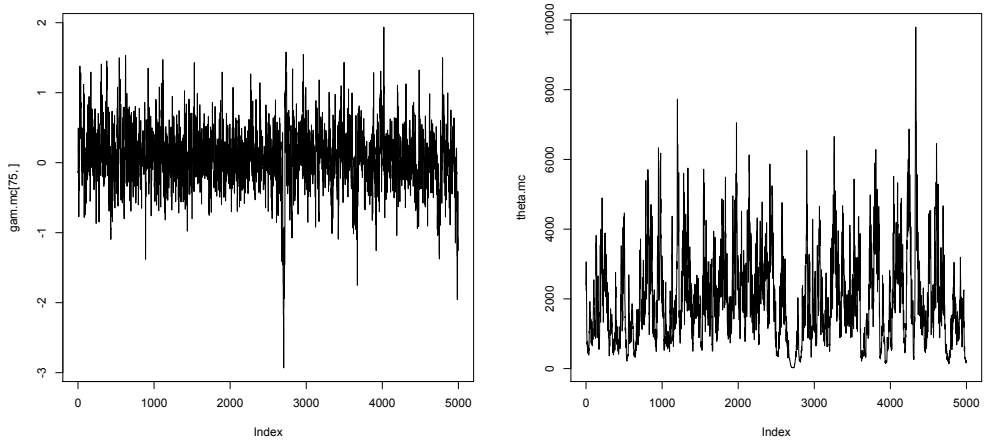


Figure 3: Examples of trace plots of MCMC method in simulation study.

	Mean	2.5% quantile	97.5% quantile
p_0	0.4026	0.3723	0.4380
β_1	0.0106	0.0040	0.0173
β_2	-0.5704	-0.9942	-0.1487
$\theta_1^{f_1}$	-0.1722	-1.2531	0.7215
$\theta_2^{f_1}$	-0.0053	-0.8922	0.6588
$\theta_3^{f_1}$	0.1780	-0.6773	1.0113
$\theta_1^{f_2}$	-1.8480	-2.1333	-1.5554
$\theta_2^{f_2}$	0.1351	0.1214	0.1481

Table 2: Posterior means and quantiles of the parameters of interest. Note that the θ's are the hyperparameters in the univariate and bivarate SDE priors.

target species in the sample is then counted.

The example in this section concerns data from a 1992 mackerel egg survey. Figure 4a shows the locations where eggs were sampled, and the egg counts found there. Besides longitude (`lon`) and latitude (`lat`), some other predictors of egg abundance are used, e.g., `salinity`; water temperature at a depth of 20 meters, `temp.20m`; and distance from the 200 meters seabed contour, `c.dist`. The latter predictor reflects the biologists' belief that the fish like to spawn near the edge of the continental shelf, conventionally considered to end at a seabed depth of 200 meters. The mackerel eggs caught in the net at each sampling location were counted, to yield the response variable `egg.count`. The whole data set is available in the `gamair` R package.

Wood (2006) assumed that the egg counts follow Poisson distributions and modeled their mean functions using generalized additive models (GAMs). However, there are many zero counts (more than 40%), suggesting that using a zero-inflated Poisson (ZIP)

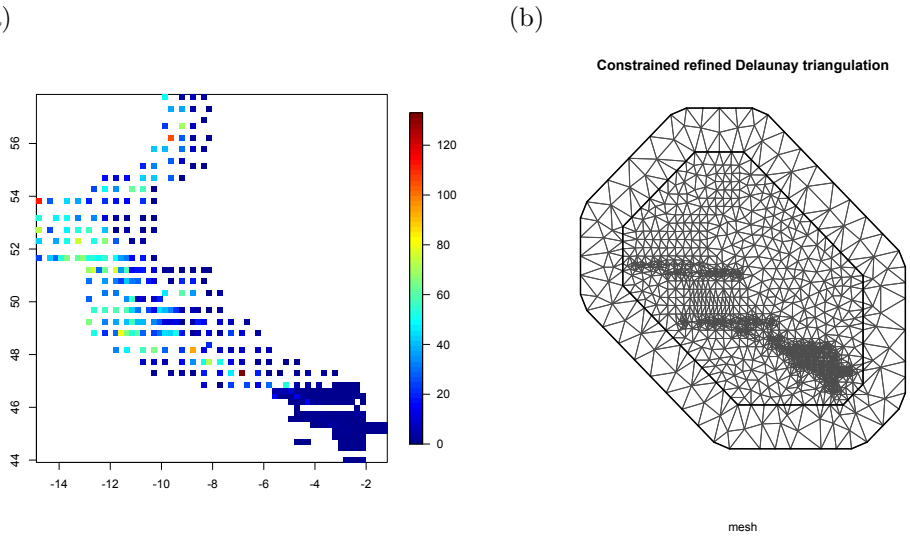


Figure 4: Panel (a) shows the mackerel egg counts of sea surface as assessed by net samples in the 1992 Mackerel egg survey. Panel (b) shows the triangular mesh built for bivariate SDE process.

model may be more appropriate here. The ZIP model has a mixture form as follows

$$\text{Prob}(Y = y_i) = p_0 1_{[y_i=0]} + (1 - p_0) \times \text{Poisson}(y_i \,|\, y_i > 0), \quad i = 1, \ldots, n, \tag{20}$$

where p_0 is the unknown probability of y_i being zero, and a Poisson distribution is assumed when y_i is positive. The mean function μ_i of the Poisson distribution is modeled as

$$\log(\mu_i) = \log(\text{net.area}_i) + \text{salinity}_i * \beta_1 + \text{c.dist}_i * \beta_2 + f_1(\text{temp.20m}_i) + f_2(\text{lon}_i, \text{lat}_i)$$

where `net.area` (area of each net) is used as an offset, β_1 and β_2 respectively represent the linear effects of `salinity` and `c.dist`, f_1 is the nonlinear effect of `temp.20m`, and f_2 is the spatial effect. We take diffuse Gaussian priors on the linear effects, and the proposed SDE priors on the nonlinear and spatial effects. More specifically, we use a quadratic B-spline basis with $m = 3$ knots to model the adaptive smoothing function in the univariate SDE prior. The bivariate SDE field is constructed on a fine triangular mesh (Figure 4b) and we let its variance depend on the longitude to achieve nonstationarity for the spatial effect. To estimate p_0, we first let $p_0 = e^\theta/(1 + e^\theta)$ and then take a Gaussian prior on θ. Table 2 shows the parameter estimates of interest. As we can see, both linear effects are statistically significant and influence the egg densities in opposite ways. None of the θ's in the univariate SDE prior are significant, indicating that adaptive smoothing may not be necessary for the nonlinear effect. It can also be seen by the slowly-varying fitted curve shown in Figure 5a. However, the θ's of the bivariate SDE prior are both significant, meaning that the nonstationarity does exist in the spatial effect. In Figure 5b we plot the variance at each location against

its longitude, where an exponentially decreasing relationship is found. The posterior mean and standard deviation to the spatial effect are presented in Figure 5c and 5d, respectively. Notice that the densities are generally high on the western boundary of the survey area, and the standard deviations are especially large around Latitude 52 and Longitude -14.

One purpose of this modeling is assessment of the total stock of eggs, which needs predictions from the model. Therefore a simple map of predicted densities is useful. The data frame `mackp` contains the model covariates on a regular grid over the survey area, as well as an indexing column indicating which grid square the covariates belong to. A naive prediction method is made by the projection of the sum of the linear predictor components and applying exponentiation to it (because we use the log link). A better alternative is the joint prediction with the estimation process, which can be efficiently implemented with INLA. The predicted egg counts and their standard deviations are shown in Figure 5e and 5f.

References

Abramovich, F. and Steinberg, D. M. (1996). Improved inference in nonparametric regression using L_k-smoothing splines. *Journal of Statistical Planning and Inference* **49**, 327–341. 398

Baladandayuthapani, V., Mallick, B. K. and Carroll, R. J. (2005). Spatially adaptive Bayesian penalized regression splines (P-splines). *Journal of Computational and Graphical Statistics* **14**, 378–394. 398

Brezger, A. and Lang, S. (2006). Generalized structured additive regression based on Bayesian P-splines. *Computational Statistics and Data Analysis* **50**, 967–991. 398

Crainiceanu, C., Ruppert, D., Carroll, R., Adarsh, J. and Goodner, B. (2007). Spatially adaptive Penalized splines with heteroscedastic errors. *Journal of Computational and Graphical Statistics* **16**, 265–288. 398

Cummins, D. J., Filloon, T. G. and Nychka, D. (2001). Confidence intervals for nonparametric curve estimates: Toward more uniform pointwise coverage. *Journal of the American Statistical Association* **96**, 233–246. 398

Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables* (W. Schempp and K. Zeller, eds.), volume 571 of *Lecture Notes in Mathematics*, 85–100, Springer Berlin / Heidelberg, 10.1007/BFb0086566. 405

Eilers, P. and Marx, B. (1996). Flexible smoothing with B-splines and penalties (with discussion). *Statistical Science* **11**, 89–121. 398

Eilers, P. H. C. and Marx, B. D. (2010). Splines, knots, and penalties. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**, 637–653. 399

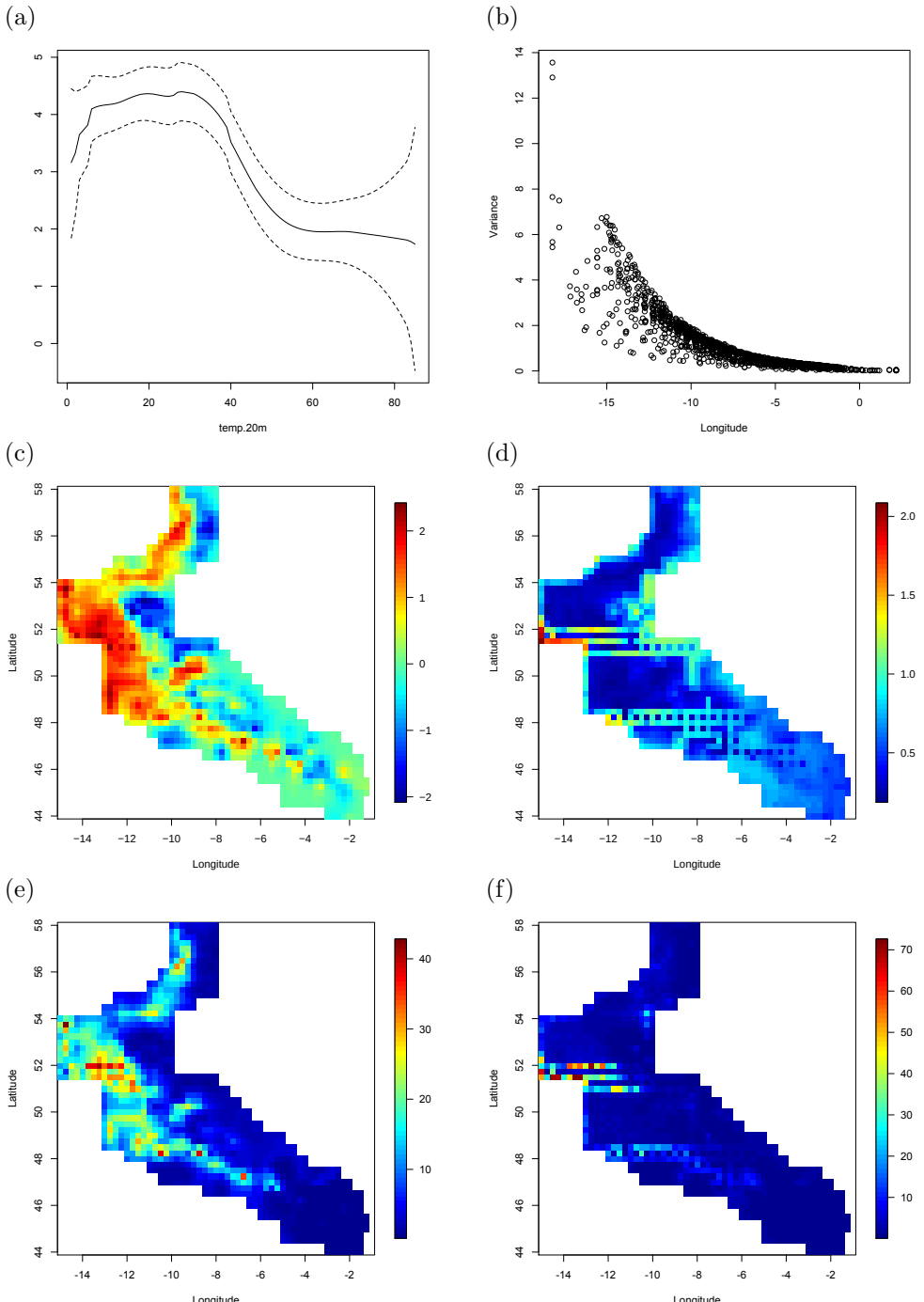


Figure 5: Modeling results for Mackerel egg data: (a) fitted curve for nonlinear effect; (b) scatterplot of spatial variance and longitude; (c) & (d) posterior mean and standard deviation to the spatial effect; (e) & (f) predicted egg counts and their standard deviations.

Eubank, R. L. (1999). *Nonparametric Regression and Spline Smoothing*. Marcel Dekker Inc. 398

Fahrmeir, L. and Knorr-Held, L. (2000). Dynamic and semiparametric models. In *Smoothing and regression: approaches, computation, and application* (M. G. Schimek, ed.), 513–544, New York: Wiley. 399

Fahrmeir, L. and Lang, S. (2001). Bayesian inference for generalized additive mixed models based on Markov random field priors. *Journal of the Royal Statistical Society, Series C: Applied Statistics* **50**, 201–220. 399

Fahrmeir, L. and Tutz, G. (2001). *Multivariate Statistical Modeling based on Generalized Linear Models*. Berlin: Springer. 408

Fahrmeir, L. and Wagenpfeil, S. (1996). Smoothing hazard functions and time-varying effects in discrete duration and competing risks models. *Journal of the American Statistical Association* **91**, 1584–1594. 399

Green, P. J. and Silverman, B. W. (1994). *Nonparametric Regression and Generalized Linear Models: a Roughness Penalty Approach*. Chapman & Hall Ltd. 401

Gu, C. (2002). *Smoothing Spline ANOVA Models*. Springer-Verlag Inc, New York. 398, 405

Kimeldorf, G. S. and Wahba, G. (1970). A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics* **41**, 495–502. 400

Krivobokova, T., Crainiceanu, C. M. and Kauermann, G. (2008). Fast Adaptive Penalized Splines. *Journal of Computational and Graphical Statistics* **17**, 1–20. 398, 411

Lang, S. and Brezger, A. (2004). Bayesian P-splines. *Journal of Computational and Graphical Statistics* **13**, 183–212. 398

Lang, S., Fronk, E. M. and Fahrmeir, L. (2002). Function estimation with locally adaptive dynamic models. *Computational Statistics* **17**, 479–499. 399

Lindgren, F. and Rue, H. (2008). On the second-order random walk model for irregular locations. *Scandinavian Journal of Statistics* **35**, 691–700. 399, 400

Lindgren, F., Rue, H. and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73**, 423–498. 405, 406

Luo, Z. and Wahba, G. (1997). Hybrid adaptive splines. *Journal of the American Statistical Association* **92**, 107–116. 398

O'Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems. *Statistical Science* **1**, 502–527. 398

Pintore, A., Speckman, P. L. and Holmes, C. C. (2006). Spatially adaptive smoothing splines. *Biometrika* **93**, 113–125. 398, 403

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*, volume 104 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London. 399, 400, 407, 408, 409

Rue, H., Martino, S. and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models using integrated nested Laplace approximations (with discussion). *Journal of the Royal Statistical Society, Series B: Statistical Methodology* **71**, 319–392. 399, 409

Ruppert, D. and Carroll, R. J. (2000). Spatially-adaptive penalties for spline fitting. *Australian & New Zealand Journal of Statistics* **42**, 205–223. 398

Ruppert, D., Wand, M. and Carroll, R. (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge. 398

Scheipl, F. and Kneib, T. (2009). Locally adaptive Bayesian P-splines with a normal-exponential-gamma prior. *Computational Statistics and Data Analysis* **53**, 3533–3552. 398

Simpson, D., Helton, K. and Lindgren, F. (2012). On the connection between O'Sullivan splines, continuous random walk models, and smoothing splines. Technical report, Norwegian University of Science and Technology. 398, 400, 402

Speckman, P. L. and Sun, D. (2003). Fully Bayesian spline smoothing and intrinsic autoregressive priors. *Biometrika* **90**, 289–302. 398, 399, 400

Staniswalis, J. G. (1989). Local bandwidth selection for kernel estimates. *Journal of the American Statistical Association* **84**, 284–288. 398

Staniswalis, J. G. and Yandell, B. S. (1992). Locally adaptive smoothing splines. *Journal of Statistical Computation and Simulation* **43**, 45–53. 398

Wahba, G. (1978). Improper priors, spline smoothing and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society, Series B: Statistical Methodology* **40**, 364–372. 400, 402

Wahba, G. (1990). *Spline Models for Observational Data*. SIAM [Society for Industrial and Applied Mathematics], Philadelphia. 397, 398, 406

Walsh, J. (1986). An introduction to stochastic partial differential equations. In *École d'Été de Probabilités de Saint Flour XIV - 1984* (R. Carmona, H. Kesten and J. Walsh, eds.), volume 1180 of *Lecture Notes in Mathematics*, 265–439, Springer Berlin / Heidelberg, 10.1007/BFb0074920. 401

Wand, M. P. and Ormerod, J. T. (2008). On semiparametric regression with O'Sullivan penalized splines. *Australian and New Zealand Journal of Statistics* **50**, 179–198. 398

Wecker, W. and Ansley, C. (1983). The signal extraction approach to nonlinear regression and spline smoothing. *Journal of the American Statistical Association* **78**, 81–89. 400

Wood, S. (2006). *Generalized additive models: an introduction with R*. CRC Press. 413

Yue, Y., Speckman, P. and Sun, D. (2012). Priors for Bayesian adaptive spline smoothing. *Annals of the Institute of Statistical Mathematics* **64**, 577–613, 10.1007/s10463-010-0321-6. 399, 400, 405

Yue, Y. and Speckman, P. L. (2010). Nonstationary spatial Gaussian Markov random fields. *Journal of Computational and Graphical Statistics* **19**, 96–116. 405, 406

Appendix

This section includes detailed proofs for the weak solutions of both adaptive SDEs.

Adaptive SDE I

Using basis expansion (6), the adaptive SDE (13) becomes a linear equation system, whose left hand side can be written as $\boldsymbol{H}_\lambda \boldsymbol{w}$. We here show how to derive the non-zero entries of the matrix $\boldsymbol{H}_\lambda$. Using integration-by-parts, we have the $[i,j]$th entry of $\boldsymbol{H}_\lambda$ as

$$\begin{aligned}
\boldsymbol{H}_\lambda[i,j] &= \left\langle \psi_i(s), \lambda(s)\psi_j''(s) \right\rangle \\
&= \int_{s_1}^{s_n} \lambda(s)\psi_i(s)\psi_j''(s)ds \\
&= \lambda(s)\psi_i(s)\psi_j'(s)\big|_{s_1}^{s_n} - \int_{s_1}^{s_n} [\lambda(s)\psi_i(s)]'\psi_j'(s)ds \\
&= \lambda(s)\psi_i(s)\psi_j'(s)\big|_{s_1}^{s_n} - \int_{s_1}^{s_n} \lambda'(s)\psi_i(s)\psi_j'(s)ds - \int_{s_1}^{s_n} \lambda(s)\psi_i'(s)\psi_j'(s)ds.
\end{aligned}$$

Since basis element ψ_i only overlaps for neighboring locations, the nonzero entries in the ith row of $\boldsymbol{H}_\lambda$ are $\boldsymbol{H}_\lambda[i,i-1]$, $\boldsymbol{H}_\lambda[i,i]$ and $\boldsymbol{H}_\lambda[i,i+1]$ for $i = 2,\ldots,n-1$. Specifically, we have

$$\begin{aligned}
\boldsymbol{H}[i,i-1] &= -\int_{s_{i-1}}^{s_i} \lambda'(s)\psi_i(s)\psi_{i-1}'(s)ds - \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)\psi_{i-1}'(s)ds \\
&= -\lambda(s)\psi_i(s)\psi_{i-1}'(s)\big|_{s_{i-1}}^{s_i} + \int_{s_{i-1}}^{s_i} \lambda(s)[\psi_i(s)\psi_{i-1}'(s)]'ds \\
&\quad - \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)\psi_{i-1}'(s)ds \\
&= -\lambda(s)\psi_i(s)\psi_{i-1}'(s)\big|_{s_{i-1}}^{s_i} + \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)\psi_{i-1}'(s)ds \\
&\quad - \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)\psi_{i-1}'(s)ds \\
&= -\lambda(s_i)\psi_i(s_i)\psi_{i-1}'(s_i) + \lambda(s_{i-1})\psi_i(s_{i-1})\psi_{i-1}'(s_{i-1}) \\
&= \lambda(s_i)/h_{i-1}.
\end{aligned}$$

Note that $\psi_{i-1}'(s)$ is constant between s_{i-1} and s_i, and thus we have $[\psi_i(s)\psi_{i-1}'(s)]' =$

$\psi_i'(s)\psi_{i-1}'(s)$. Similarly, we have

$$\begin{aligned}
\boldsymbol{H}_\lambda[i,i] &= -\int_{s_{i-1}}^{s_{i+1}} \lambda'(s)\psi_i(s)\psi_i'(s)ds - \int_{s_{i-1}}^{s_{i+1}} \lambda(s)\psi_i'(s)^2 ds \\
&= -\int_{s_{i-1}}^{s_i} \lambda'(s)\psi_i(s)\psi_i'(s)ds - \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)^2 ds \\
&\quad -\int_{s_i}^{s_{i+1}} \lambda'(s)\psi_i(s)\psi_i'(s)ds - \int_{s_i}^{s_{i+1}} \lambda(s)\psi_i'(s)^2 ds \\
&= -\lambda(s)\psi_i(s)\psi_i'(s)|_{s_{i-1}}^{s_i} + \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)^2 ds - \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)^2 ds \\
&\quad -\lambda(s)\psi_i(s)\psi_i'(s)|_{s_i}^{s_{i+1}} + \int_{s_i}^{s_{i+1}} \lambda(s)\psi_i'(s)^2 ds - \int_{s_i}^{s_{i+1}} \lambda(s)\psi_i'(s)^2 ds \\
&= -\lambda(s_i)\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right), \\
\boldsymbol{H}_\lambda[i,i+1] &= -\int_{s_i}^{s_{i+1}} \lambda'(s)\psi_i(s)\psi_{i+1}'(s)ds - \int_{s_i}^{s_{i+1}} \lambda(s)\psi_i'(s)\psi_{i+1}'(s)ds \\
&= -\lambda(s)\psi_i(s)\psi_{i+1}'(s)|_{s_i}^{s_{i+1}} + \int_{s_i}^{s_{i+1}} \lambda(s)\psi_i'(s)\psi_{i+1}'(s)ds \\
&\quad -\int_{s_i}^{s_{i+1}} \lambda(s)\psi_i'(s)\psi_{i+1}'(s)ds \\
&= -\lambda(s_{i+1})\psi_i(s_{i+1})\psi_{i+1}'(s_{i+1}) + \lambda(s_i)\psi_i(s_i)\psi_{i+1}'(s_i) \\
&= \lambda(s_i)/h_i.
\end{aligned}$$

For the first and last row of $\boldsymbol{H}_\lambda$, the (possible) nonzero entries are $\boldsymbol{H}_\lambda[1,1]$, $\boldsymbol{H}_\lambda[1,2]$, $\boldsymbol{H}_\lambda[n,n-1]$ and $\boldsymbol{H}_\lambda[n,n]$, which happen to be zeroes due to the intrinsic condition. We here only show the derivation of $\boldsymbol{H}_\lambda[1,1]$ and $\boldsymbol{H}_\lambda[1,2]$, and the other entries can be obtained similarly. We have

$$\begin{aligned}
\boldsymbol{H}_\lambda[1,1] &= \lambda(s)\psi_1(s)\psi_1'(s)|_{s_1}^{s_n} - \int_{s_1}^{s_2} \lambda'(s)\psi_1(s)\psi_1'(s)ds - \int_{s_1}^{s_2} \lambda(s)\psi_1'(s)^2 ds \\
&= -\lambda(s_1)\psi_1(s_1)\psi_1'(s_1) - \lambda(s)\psi_1(s)\psi_1'(s)|_{s_1}^{s_2} \\
&= -\lambda(s_1)\psi_1(s_1)\psi_1'(s_1) + \lambda(s_1)\psi_1(s_1)\psi_1'(s_1) \\
&= 0, \\
\boldsymbol{H}_\lambda[1,2] &= \lambda(s)\psi_1(s)\psi_2'(s)|_{s_1}^{s_n} - \int_{s_1}^{s_2} \lambda'(s)\psi_1(s)\psi_2'(s)ds - \int_{s_1}^{s_2} \lambda(s)\psi_1'(s)\psi_2'(s)ds \\
&= -\lambda(s_1)\psi_1(s_1)\psi_2'(s_1) - \lambda(s)\psi_1(s)\psi_2'(s)|_{s_1}^{s_2} \\
&= -\lambda(s_1)\psi_1(s_1)\psi_2'(s_1) + \lambda(s_1)\psi_1(s_1)\psi_2'(s_1) \\
&= 0.
\end{aligned}$$

Finally, we can easily see that $\boldsymbol{H}_\lambda = \boldsymbol{\Lambda}\boldsymbol{H}$, where $\boldsymbol{\Lambda}$ is the diagonal matrix of $\lambda(\cdot)$'s and $\boldsymbol{H}$ is the tridiagonal matrix defined as in (9).

Adaptive SDE II

Letting $\tilde{f} = \lambda(s)f(s)$, the left hand side of (15) can be written as

$$\begin{aligned}\left\langle \psi_i(s), \tilde{f}''(s) \right\rangle &= \int \psi_i(s)\tilde{f}''(s)ds \\ &= \psi_i(s)\tilde{f}'(s)\Big|_{s_1}^{s_n} - \int \psi_i'(s)\tilde{f}'(s)ds.\end{aligned}$$

Using basis expansion (6), the adaptive SDE (15) becomes a linear equation system, whose left hand side can be written as $\boldsymbol{H}_\lambda \boldsymbol{w}$. We then have the $[i,j]$th entry of $\boldsymbol{H}_\lambda$ as

$$\begin{aligned}\boldsymbol{H}_\lambda[i,j] &= \lambda'(s)\psi_i(s)\psi_j(s)|_{s_1}^{s_n} + \lambda(s)\psi_i(s)\psi_j'(s)\Big|_{s_1}^{s_n} \\ &\quad - \int_{s_1}^{s_n} \lambda'(s)\psi_i'(s)\psi_j(s)ds - \int_{s_1}^{s_n} \lambda(s)\psi_i'(s)\psi_j'(s)ds.\end{aligned}$$

Since basis element ψ_i only overlaps for neighboring locations, the nonzero entries in the ith row of $\boldsymbol{H}_\lambda$ are $\boldsymbol{H}_\lambda[i,i-1]$, $\boldsymbol{H}_\lambda[i,i]$ and $\boldsymbol{H}_\lambda[i,i+1]$ for $i = 2,\ldots,n-1$. Specifically, we have

$$\begin{aligned}\boldsymbol{H}[i,i-1] &= -\int_{s_{i-1}}^{s_i} \lambda'(s)\psi_i'(s)\psi_{i-1}(s)ds - \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)\psi_{i-1}'(s)ds \\ &= -\lambda(s)\psi_i'(s)\psi_{i-1}(s)|_{s_{i-1}}^{s_i} + \int_{s_{i-1}}^{s_i} \lambda(s)[\psi_i'(s)\psi_{i-1}(s)]'ds \\ &\quad - \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)\psi_{i-1}'(s)ds \\ &= -\lambda(s)\psi_i'(s)\psi_{i-1}(s)|_{s_{i-1}}^{s_i} + \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)\psi_{i-1}'(s)ds \\ &\quad - \int_{s_{i-1}}^{s_i} \lambda(s)\psi_i'(s)\psi_{i-1}'(s)ds \\ &= -\lambda(s_i)\psi_i'(s_i)\psi_{i-1}(s_i) + \lambda(s_{i-1})\psi_i'(s_{i-1})\psi_{i-1}(s_{i-1}) \\ &= \lambda(s_{i-1})/h_{i-1}.\end{aligned}$$

Note that $\psi_i'(s)$ is constant between s_{i-1} and s_i, and thus we have $[\psi_i'(s)\psi_{i-1}(s)]' = \psi_i'(s)\psi_{i-1}'(s)$. Similarly, we have

$$\boldsymbol{H}_\lambda[i,i] = -\int_{s_{i-1}}^{s_{i+1}} \lambda'(s)\psi_i'(s)\psi_i(s)ds - \int_{s_{i-1}}^{s_{i+1}} \lambda(s)\psi_i'(s)^2 ds = -\lambda(s_i)\left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right),$$

which is the same as in the previous case, and

$$\begin{aligned}
\boldsymbol{H}_\lambda[i,i+1] &= -\int_{s_i}^{s_{i+1}} \lambda'(s)\psi_i'(s)\psi_{i+1}(s)ds - \int_{s_i}^{s_{i+1}} \lambda(s)\psi_i'(s)\psi_{i+1}'(s)ds \\
&= -\lambda(s)\psi_i'(s)\psi_{i+1}(s)|_{s_i}^{s_{i+1}} + \int_{s_i}^{s_{i+1}} \lambda(s)\psi_i'(s)\psi_{i+1}'(s)ds \\
&\quad - \int_{s_i}^{s_{i+1}} \lambda(s)\psi_i'(s)\psi_{i+1}'(s)ds \\
&= -\lambda(s_{i+1})\psi_i'(s_{i+1})\psi_{i+1}(s_{i+1}) + \lambda(s_i)\psi_i'(s_i)\psi_{i+1}(s_i) \\
&= \lambda(s_{i+1})/h_i.
\end{aligned}$$

For the first and last row of $\boldsymbol{H}_\lambda$, the (possible) nonzero entries are $\boldsymbol{H}_\lambda[1,1]$, $\boldsymbol{H}_\lambda[1,2]$, $\boldsymbol{H}_\lambda[n,n-1]$ and $\boldsymbol{H}_\lambda[n,n]$, of which the first two entries can be derived as

$$\begin{aligned}
\boldsymbol{H}_\lambda[1,1] &= \lambda'(s)\psi_1^2(s)|_{s_1}^{s_n} + \lambda(s)\psi_1(s)\psi_1'(s)|_{s_1}^{s_n} \\
&\quad - \int_{s_1}^{s_2} \lambda'(s)\psi_1'(s)\psi_1(s)ds - \int_{s_1}^{s_2} \lambda(s)\psi_1'(s)^2 ds \\
&= -\lambda'(s_1)\psi_1^2(s_1) - \lambda(s_1)\psi_1'(s_1)\psi_1(s_1) - \lambda(s_1)\psi_1'(s)\psi_1(s)|_{s_1}^{s_2} \\
&= -\lambda'(s_1)\psi_1^2(s_1) - \lambda(s_1)\psi_1'(s_1)\psi_1(s_1) + \lambda(s_1)\psi_1'(s_1)\psi_1(s_1) \\
&= -\lambda'(s_1), \\
\boldsymbol{H}_\lambda[1,2] &= \lambda'(s)\psi_1(s)\psi_2(s)|_{s_1}^{s_n} + \lambda(s)\psi_1(s)\psi_2'(s)|_{s_1}^{s_n} \\
&\quad - \int_{s_1}^{s_2} \lambda'(s)\psi_1'(s)\psi_2(s)ds - \int_{s_1}^{s_2} \lambda(s)\psi_1'(s)\psi_2'(s)ds \\
&= -\lambda'(s_1)\psi_1(s_1)\psi_2(s_1) - \lambda(s_1)\psi_1(s_1)\psi_2'(s_1) \\
&\quad -\lambda(s_2)\psi_1'(s_2)\psi_2(s_2) + \lambda(s_1)\psi_1'(s_1)\psi_2(s_1) \\
&= -\lambda(s_1)/h_1 + \lambda(s_2)/h_1.
\end{aligned}$$

Similarly, the last two entries are given by

$$\boldsymbol{H}_\lambda[n-1,n] = \lambda(s_{n-1})/h_{n-1} - \lambda(s_n)/h_{n-1} \quad \text{and} \quad \boldsymbol{H}_\lambda[n,n] = \lambda'(s_n).$$

These four entries can be viewed as (at least approximately) the derivatives of $\lambda(s)$ at the boundary points. To be consistent with the previous case, we assume the Neumann boundary condition: $\lambda'(s_1) = \lambda'(s_n) = 0$, to make the entries be zeroes. Then, we can easily see that $\boldsymbol{H}_\lambda = \boldsymbol{H}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is the diagonal matrix of $\lambda(\cdot)$'s and $\boldsymbol{H}$ is the matrix defined as in (9).